

REMARKS

Claim rejections under 35 USC 103

Claims 1-5, 8-9, 11-12, and 14-18 have been rejected under 35 USC 103(a) as being unpatentable over Johnson (5,796,972) in view of Jim Handy, The Cache Memory Book, 2<sup>nd</sup> edition, which is hereinafter referred to as Handy, and further in view of Dockser (6,006,030). Claims 7, 10, and 13 have been rejected under 35 USC 103(a) as being unpatentable over Johnson in view of Handy, further in view of Dockser, and further again in view of “The PowerPC Architecture” reference. Claim 19 has been rejected under 35 USC 103(a) as being unpatentable over Johnson in view of Handy, further in view of Dockser, and further again in view of IBM Technical Disclosure Bulletin, vol. 37, no. 03. Claims 1, 8, 11, 14, 16, and 18 are independent claims, from which the other claims ultimately depend. Applicant respectfully traverses this rejection as to the independent claims as have been amended, such that all the pending claims rejected on this basis are patentable.

Applicant respectfully submits that the claimed invention is not rendered *prima facie* obvious over Johnson in view of Handy, and further in view of Dockser, for two reasons. First, modifying Johnson in view of Handy would not work, and/or would require a substantial reconstruction or redesign of Johnson to work, which is impermissible – and there is indeed no motivation to performing this modification in the first place. Second, there is no motivation to modify Johnson in view of Handy further in view of Dockser, insofar as the Examiner’s stated motivations to modify the combination of Johnson and Handy in view of Dockser are actually not real motivations at all. Applicant initially discusses what Johnson discloses, then looks at the modifications needed by Johnson in view of Handy (and why these would not work and/or would require a substantial reconstruction or redesign of Johnson to work), and finally looks at why there is no motivation to modify the combination of Johnson and Handy in view of Dockser.

*The Johnson reference*

First, as noted by the Examiner, FIG. 8 of Johnson operates to decode an instruction. An identification of the instruction is stored in the ID RAM 292. This identification is provided to the UCODE ROM 294, which stores a standard output for the instruction, and to the PATCH ROM 296, which stores an alternative output for the instruction. The multiplexer 298 determines which of these two outputs are to actually be used for the instruction. As stated in Johnson itself:

[A]n instruction address [is provided] to the microcode ID RAM 292 . . . . The ID RAM 292 decodes the instruction address by using a table look-up algorithm, and provides a decoded address to the microcode RAM 296 and microcode ROM 294 via interface 302. The microcode RAM 296 and microcode ROM 294 read the corresponding address locations, and each provide[] a corresponding microcode instruction on interfaces 295 and 293 respectively. The multiplexer 298 selects which of the instructions to execute. If the current instruction provided on interface 302 does not require a microcode patch, the multiplexer 298 selects the output of the microcode ROM 294. If, however, the current instruction requires a microcode patch, the multiplexer selects the output of the microcode RAM 296.

(Col. 12, ll. 4-18) Thus, in this respect, Johnson operates quite well as to provide one of two outputs for a given instruction – either the standard output in the ROM 294, or a “patch” output in the RAM 296.

It is informative to see how Johnson decides which output to use – the standard output in the ROM 294, or the alternative output in the RAM 296 – for a given instruction. This is described in Johnson in relation to FIG. 10:

Each instruction may provide, or may be decoded to provide, an address to ID RAM 352 to select a corresponding instruction entry. In the illustrative embodiment, an INST-J provides an address to select the corresponding INST-J entry 354 in the ID RAM 352. The INST-J entry in the ID RAM 352 includes . . . a CS RAM SELECT bit of “1”. The CS RAM SELECT bit indicates that a patch instruction is required for the INST-J instruction, and is provided to multiplexer 364.

The INST-J address is provided to both the microcode ROM 360 and the microcode RAM 356, as shown. The microcode ROM 360 reads the contents at the INST-J address, or in this case, the microcode for INST-J, as shown at 380.

Likewise, the microcode RAM 356 reads the contents of the INST-J address, or in this case, the microcode for INST-J, as shown at 376. . . .

As indicated above, the CS RAM SELECT bit is provided to the multiplexer 364 via interface 362. Since the CS RAM SELECT bit is set to “1”, the multiplexer 364 selects the output of the microcode RAM 356, and provides the microcode patch instruction to the microcode instruction output 366.

(Col. 13, l. 60, through col. 14, l. 17) Thus, the decoded instruction includes a CS RAM SELECT bit. This bit is passed by the microcode RAM 356 to the multiplexer, and where this bit is equal to one, the multiplexer selects the alternative output provided by the RAM 356 instead of the output provided by the ROM 360.

*Modifying Johnson in view of Handy*

Now, let us look at how the Examiner would modify Johnson in view of Handy. The Examiner would implement the ID RAM 292 of FIG. 8 of Johnson as a CAM memory of Handy, which the Examiner has equated with a comparator. More specifically, this is what the Examiner would do:

Instead of providing an address as an input, a data input is provided, which is then compared against a stored value. When a match is present, the corresponding stored data is output. Implementing the ID RAM as a CAM memory instead of an index memory would cause the identifying information to be provided to a comparator and some of the decoded identifying information would still be provided to the UCODE ROM 294. The output of the comparator . . . would consist of a CS RAM SEL bit, which results from a match and is input to the multiplexer to select from which memory, the predetermined (ROM 294) or the alternative (RAM 296), the output is to be selected from.

(Office action, p. 4) Let us parse this a little bit. First, Johnson simply says that the “ID RAM 292 decodes the instruction address by using a table look-up algorithm, and provides a decoded address to the microcode RAM 296 and microcode ROM 294.” (Col. 12, ll. 6-8) The Examiner, though, says that “[i]nstead of providing an address as an input, a data input is provided, which is

then compared against a stored value,” such that “[w]hen a match is present, the corresponding stored data is output.”

This entails a pretty big modification of Johnson. The Examiner is redesigning the ID RAM 292 to instead be a CAM memory/comparator, and thus now requires that instead of an address being provided as an input, a data input has to be provided – and there has to be a stored value in this CAM memory/comparator against which it is being compared. But Johnson is pretty specific as to what is input into the ID RAM 292 – simply an instruction address. Where would this “data input” come from that the Examiner would have to be input into the CAM memory/comparator? The pre-decode block 290 provides an instruction address, not a data input, so somehow the Examiner would have to redesign Johnson so that a data input would be provided. Just here, the Examiner is requiring a major reconstruction and redesign of Johnson. By replacing the ID RAM 292 with a CAM memory/comparator, the pre-decode block 290 would have to be modified to provide a data input instead of an instruction address, and this only begs the question as to how the pre-decode block would get this data input in the first place, and so on.

Then, the Examiner says you have to compare this data input against a stored value. But where does this stored value come from? The ID RAM 292 only receives one input, the instruction address from the pre-decode block 290. Would the Examiner require further reconstruction and redesign of Johnson so that it receives an additional input to receive the value against which to compare the instruction address? Where would this additional input come from? Would somehow the pre-decode block 290 provide this additional input, too? Would the CAM memory/comparator have to store a value against which the data input compared for every different type of data input that could be provided? Wouldn't this require a lot of extra storage space? But, Johnson says that “it is often desirable to . . . minimize the use and size of RAMs.” (Col. 12, ll. 18-22)

In short, it is difficult to imagine how one of ordinary skill within the art could simply “pop out” the ID RAM 292 and “pop in” the CAM memory/comparator of Handy without substantially redesigning and reconstructing how Johnson works. The Examiner has pointed to two changes that would already have to be made. First, instead of an instruction address provided as input to the ID RAM 292, some ambiguous “data input” would have to provide to the CAM memory/comparator. Second, this “data input” would have to be compared against a “stored value” within the CAM memory/comparator. Applicant respectfully submits that the Examiner does not appreciate how these changes would require a substantial redesign and reconstruction as to how Johnson works. The pre-decode block 290 would have to be changed to provide this “data input” instead of an instruction address. Other components upstream would likewise have to be modified. An entirely new input line would have to be added to provide the CAM memory/comparator with the “stored value” against which it compares this “data input.” And which component of Johnson would provide this value to be stored in the CAM memory/comparator.

We then get to the fact that there really is no motivation to modify Johnson in view of Handy in this substantial way as the Examiner would have us do. Let us look at the Examiner’s stated motivations for modifying Johnson in view of Handy.

It would have been obvious to one of ordinary skill in the art to replace the indexable memory of Johnson with the CAM memory of Handy since [1] CAM memories are widely known in the art as alternatives to indexable memories and [2] allows any entry to point to any location and have parallel access to each comparator [3] so there is no hindrance on performance. Furthermore, [4], a CAM memory would logically perform the same operation the indexable memory is performing. An identifying data portion is input to the memory, and the associated data is retrieved, in both the indexable memory and the CAM memory.

(Office action, p. 4) Let us look at these four stated motivations. First, that CAM memories are widely known in the art as alternatives to indexable memories is not a motivation to replace the ID RAM of Johnson with the CAM memory of Handy; just because a modification *can* be made does not mean that it is *desirable* to make this modification.

Second, that CAM memories allow any entry to point to any location and have parallel access to each comparator is a bit inscrutable of a motivation. How is this different than the ID RAM of Johnson, and why is this “better” than what the ID RAM of Johnson does? The Examiner does not say. Furthermore, having “parallel access to each comparator” is itself circular reasoning. Johnson does not have a comparator in the first place; the entire point of introducing Handy is to add such a comparator. To say that CAM memory allows parallel access to each comparator is really to say simply that “CAM memory allows parallel access to each CAM memory,” since the Examiner has equated CAM memory with a comparator. (By comparison, ID RAM is by definition “random access” memory, and therefore allows random access to its stored contents, and apparently CAM memory by definition “parallel access” memory, allowing parallel access to its stored contents.)

Third, that “there is no hindrance on performance” seems to be a justification as to why CAM memories are *no worse* than the ID RAM of Johnson, not why they are better. You do not replace one thing with another so that “there is no hindrance on performance” as compared to what you are doing now. Rather, you need a motivation for actually replacing one thing with another – like a performance *benefit*, not that there will not be any performance hindrance.

Finally, the Examiner’s fourth statement, that a CAM memory would logically perform the same operation the indexable memory is performing is false on its face. As the Examiner himself already has admitted, while an address would be provided as an input to the ID RAM, a data input would have to be provided to the CAM memory, and then – because the CAM memory is a comparator – you would have to have some sort of “stored value” against which to compare the data input. This is not logically performing the same operation at all. There is a different type of input, and a different type of operation being performed – a comparison operation. Indeed, the entire reason for introducing the CAM memory of Handy into Johnson is to add a comparator, which the Examiner has admitted that Johnson does not have. So if Johnson does not have a comparator, and thus does not perform comparator functionality, how can the CAM

memory/comparator of Handy be logically equivalent to the ID RAM of Johnson, which is not a comparator and does not perform comparator functionality?

In sum, then, Johnson is not properly modified per Handy. First, as has been discussed, modifying Johnson in view of Handy would require a substantial reconstruction and redesign of Johnson to use a comparator/CAM memory instead of its ID RAM. Second, as has also been discussed, there is no motivation to perform this modification – just because one of ordinary skill within the art possibly *could* modify the ID RAM of Johnson to instead be a comparator/CAM memory (albeit with substantial difficulty!), the Examiner has not provided any reason as to why this is desirable. That CAM memories are “widely known in the art,” “allow parallel access to comparators” (which is not an advantage, since Johnson does not have any comparators in the first place until you already modify Johnson per Handy) and “logically perform the same operation” does not suggest the desirability of using the CAM memories instead of the ID RAM.

*Modifying Johnson and Handy in view of Dockser*

Finally, let us look at whether there is any motivation to modify the combination of Johnson and Handy in view of Dockser. Dockser is relied upon by the Examiner to provide the masking register functionality of the claimed invention. The Examiner’s stated motivation for modifying Johnson and Handy per Dockser is as follows.

The advantage of using a masking register is that it allows for the relevant bits to pass through and the irrelevant bits to be ignored. One of ordinary skill in the art would have been motivated by this advantage to implement a mask register to only allow the relevant bits for a matching instruction to pass through to a comparator. Thus, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement a mask register to get rid of irrelevant bits for the advantage of making the comparison easier to detect the correct instructions.

(Office action, p. 5)

Applicant asks the Examiner one question as to this stated motivation: where are the “irrelevant bits” in the combination of Johnson and Handy that would require the further

modification per Dockser to add masking register functionality to get rid of these “irrelevant bits”? To be sure, Johnson by itself does not have any “irrelevant bits”. As has been stated above in relation to FIG. 10 of Johnson, Johnson uses a single CS RAM SELECT bit to determine whether to use the output from the ROM or the output from the RAM. There is just a single bit that the multiplexer uses. There are no “irrelevant bits” that have to be masked out. Therefore, there is no reason to modify Johnson in view of Dockser.

But, the combination of Johnson and Handy may be another story. The Examiner has replaced the perfectly functioning ID RAM of Johnson with the CAM memory of Handy, which apparently performs some sort of comparator functionality. The Examiner would have us replace the address input into the ID RAM with some sort of “data input” which is then compared against some sort of “stored value.” Now, the Examiner seems to suggest that this data input would have “irrelevant bits” that would have to be masked out in order for Johnson in view of Handy to even optimally work.

Again, this is a lot of redesign and reconstruction. Applicant asks that the Examiner step back for a moment and consider what he is requesting us to do. Johnson, without any “irrelevant bits” perfectly fine determines whether a ROM output or a RAM output should be used for a given instruction. Now, to yield the claimed invention, the Examiner first has to replace the ID RAM of Johnson with a CAM memory/comparator (since the claimed invention utilizes a comparator, such that if you are impermissibly going to use the claimed invention as a template to piece together the prior art, you have to find a comparator in the prior art). This involves new types of “data inputs” and somehow getting “stored values” into the comparator.

But then, the Examiner informs us that the CAM memory/comparator may have problems making comparisons, due to “irrelevant bits” being present (where these irrelevant bits come from, we do not know). As such, you would have to further modify the combination of Johnson and Handy to add in the masking functionality of Dockser. However, this begs the question a little bit – if we are talking about a nebulous “data input” of the combination of Johnson and



Handy in the first place, why don't we just make that data input have only relevant bits to begin with, so that there are no irrelevant bits? That is, if we're already performing the reconstruction and redesign of Johnson to replace its address input with a completely different type of data input, let's ensure that its data input does not have any irrelevant bits in the first place.

In short, Applicant respectfully submits that the Examiner cannot have it both ways. If one of ordinary skill within the art can permissibly do the reconstruction and redesign of Johnson to replace its ID RAM with a CAM memory/comparator, then surely the reconstruction and redesign would involve not having any irrelevant bits lying around that would make the comparator's job more difficult, and that would require adding Dockser into the mix to mask out. The only reason why you would not want to do this – that is, why you would not want to have no irrelevant bits lying around – is so that you *have to* perform further modification per Dockser to adding masking functionality. But, the only reason why you would want to make things this complicated is that if you are using the claimed invention as a template to piece together the prior art to yield the claimed invention, then have to have irrelevant bits that require masking out.

In summary, Applicant submits that the Examiner is going through a Rube Goldberg-esque approach of looking at the prior art to yield the claimed invention obvious. In Johnson by itself, you have a perfectly fine way of determining whether to use a predetermined ROM output or an alternative RAM output for a given instruction. Replacing one part of Johnson to use the CAM memory/comparator of Handy would at best require substantial reconstruction and redesign of Johnson, with no real advantages. Apparently, however, performing this substantial reconstruction and redesign would have one major disadvantage, in that there would be “irrelevant bits” lying around that would make the Handy comparator's job difficult, and that would further require additional modification to use the masking functionality of Dockser. Thus, the motivation for adding Dockser to the combination of Johnson and Handy is a disincentive to combine Johnson and Handy in the first place. You are breaking something that works perfectly fine (Johnson) to add in a comparator (Handy) that has dubious performance benefits over what

Johnson already does, and which adds needless complexity and substantial reconstruction and redesign of that initial something (Johnson) – but then it turns out that this modification introduces another problem, irrelevant bits, that you need to fix, too (per Dockser). If modifying Johnson in view of Handy requires further modification per Dockser to even work, are the benefits of going through all this trouble “Really Worth It” when Johnson already works perfectly fine for its stated functionality without any modification?

In any case, for Johnson in view of Handy to be properly modified further in view of Dockser, the Examiner has to show where these alleged “irrelevant bits” are that require masking out by Dockser. Furthermore, the Examiner has to show that these “irrelevant bits” are present in Johnson alone (and which Applicant has demonstrated already that there are no such “irrelevant bits”). Otherwise, one of ordinary skill within the art would simply modify Johnson in view of Handy in the first place so that there are no “irrelevant bits” (i.e., if we’re doing major surgery on Johnson already, why not do it right?). And, if you cannot modify Johnson in view of Handy so that there are no “irrelevant bits,” then this itself is a reason not to modify Johnson in view of Handy in the first place – since Johnson by itself does not have any “irrelevant bits” that require masking by Dockser.

For all of these reasons, therefore, there is no *prima facie* obviousness of the claimed invention.

Conclusion

Applicants have made a diligent effort to place the pending claims in condition for allowance, and request that they so be allowed. However, should there remain unresolved issues that require adverse action, it is respectfully requested that the Examiner telephone Applicants' Attorney so that such issues may be resolved as expeditiously as possible. For these reasons, this application is now considered to be in condition for allowance and such action is earnestly solicited.

Respectfully Submitted,



January 10, 2007  
Date

Michael A. Dryja, Reg. No. 39,662  
Attorney/Agent for Applicant(s)

Law Offices of Michael Dryja  
704 228<sup>th</sup> Ave NE #694  
Sammamish, WA 98074  
tel: 425-427-5094

fax: 206-374-2819